



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/748,404	12/26/2000	David A. Egolf	52003199	4563

7590 03/24/2004  
DR. RUSSELL W. GUENTHNER  
BULL HN INFORMATION SYSTEMS INC.  
13430 N. BLACK CANYON HWY. B55  
PHOENIX, AZ 85029

EXAMINER

ALI, SYED J

ART UNIT	PAPER NUMBER
----------	--------------

2127

DATE MAILED: 03/24/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

## Office Action Summary

Application No.

09/748,404

Applicant(s)

EGOLF, DAVID A.

Examiner

Syed J Ali

Art Unit

2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 26 December 2000.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-21 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-21 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 26 December 2000 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)             | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)    | Paper No(s)/Mail Date. _____  |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____   | 6) <input type="checkbox"/> Other: _____                                    |

### DETAILED ACTION

1. Claims 1-21 are pending in this application.

#### *Claim Objections*

2. Claims 7, 12-17 are objected to because of the following informalities:

In line 2 of claims 7 and 17, "step (c)" should read "step (C)".

In claims 12-17, "set" should read "step" in all instances.

In claims 12-17, "subset" should read "substep" in all instances.

Appropriate correction is required.

#### *Claim Rejections - 35 USC § 102*

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1, 9-11, and 19-21 are rejected under 35 U.S.C. 102(e) as being anticipated by Jackson et al. (USPN 6,473,819) (hereinafter Jackson).

As per claim 1, Jackson teaches the invention as claimed, including a method of priority queue dispatching in a data processing system, wherein:

a critical section of code for a task is when the task has at least one of a plurality of locks locked; and

said method comprises:

A) dispatching a first task for execution at a temporary dispatching priority higher than a standard dispatching priority for that first task whenever the first task is in a critical section of code for that first task (col. 6 lines 32-37, "After receiving ownership of the lock...the computation agent eliminates interferences from interrupts by raising its priority level, i.e., IRQL, to a second priority level"), wherein the critical section of code for that first task is a result of that first task having locked a first lock without having unlocked the first lock (col. 1 lines 13-21, "Because of atomicity requirements, a thread may have to raise its priority level before entering a critical section that manipulates memory"); and

B) dispatching the first task for execution at the standard dispatching priority for that first task whenever the first task is not in the critical section of code for that first task (col. 3 lines 54-59, "the computation agent's priority level is restored to its original, i.e., first priority level, when it rejoins the queue to reacquire the lock").

As per claim 9, Jackson teaches the invention as claimed, including the method in claim 1 wherein:

the temporary dispatching priority for the first task is a system wide value (col. 2 lines 12-28, "The priority level is typically an attribute of the operating environment wherein only

interrupts above a certain priority level will be serviced. On many systems, this is referred to as the IRQL”).

As per claim 10, Jackson teaches the invention as claimed, including the method in claim 1 wherein:

the temporary dispatching priority for the first task is a value associated with the first lock when the first lock is a last lock that the first task has attempted to lock (col. 6 lines 32-37, “the computation agent eliminates interference from interrupts by raising its priority level, i.e., IRQL, to a second priority level required by the lock”).

As per claims 11 and 19-20, Jackson teaches the invention as claimed, including software stored in a computer software storage medium for performing the method of claims 1 and 9-10, respectively (Fig. 1, wherein the invention is disclosed in a computing environment including software for implementing the disclosed method).

As per claim 21, Jackson teaches the invention as claimed, including a computer readable non-volatile storage medium encoded with software for performing the method of claim 1 (Fig. 1, wherein the invention is disclosed in a computing environment including a computer readable medium containing software for implementing the disclosed method).

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 2-3, 7, 12-13, and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jackson in view of McDonald et al. (USPN 6,560,627) (hereinafter McDonald) in view of Murata (USPN 6,560,628).

As per claim 2, McDonald teaches the invention as claimed, including the following limitations not shown by Jackson, specifically the method in claim 1 wherein:

the method further comprises:

C) queuing a second task in a first FIFO queue when the second task attempts to lock the first lock and fails (col. 5 lines 30-37, "if a first task ['task A'] attempts to lock a resource, and the resource is already held by a second task ['task B'],...task A is added to the wait list"), wherein:

the FIFO queue comprises a set of tasks waiting to lock the first lock (col. 4 lines 28-37, "wait list 200 contains N task entries, labeled tasks 1 to N. Each task in the list is a task that is not currently executing and is waiting for a resource to be freed"); and

step (C) comprises:

- 1) placing the second task at an end of the first FIFO queue (col. 5 lines 30-37, "if a first task ['task A'] attempts to lock a resource, and the resource is already held by a second task ['task B'],...task A is added to the wait list").

Murata teaches the invention as claimed, including the following limitations not shown by Jackson or McDonald, specifically:

- 2) comparing the temporary dispatching priority for a third task that is ahead of the second task in the FIFO queue to the temporary dispatching priority of the second task (col. 3 lines 7-15, "the use of the priority inheritance scheme requires modification or rescheduling of the wait queue to respond to the dynamic change of the priority level", wherein the priority inheritance scheme requires an adjustment in the waiting tasks such that a first-in-first-out dispatching scheme is maintained); and
- 3) setting the temporary dispatching priority of the third task to the temporary dispatching priority of the second task when the temporary dispatching priority of the second task is determined in substep (2) to be greater than the temporary dispatching priority of the third task (col. 3 lines 7-15, "the use of the priority inheritance scheme requires modification or rescheduling of the wait queue to respond to the dynamic change of the priority level", wherein the priority inheritance scheme allows one task to inherit the priority associated with a subsequent task that has a higher priority).

It would have been obvious to one of ordinary skill in the art to combine Jackson and McDonald since the method disclosed by Jackson, while raising the priority of an executing task

to avoid a system interrupt, fails to account for the arrival of another task that may be of higher priority. Thus, the disclosure of McDonald, which allows an executing task to inherit the priority of a waiting task that is of a higher priority allows the executing task to finish its work on the shared resource, and thereby ensuring the resource is left in a consistent state upon the executing task's completion.. Further, it would have been obvious to one of ordinary skill in the art to add Murata to the combination of Jackson and McDonald since the dynamic shifting of task's priorities could result starvation for certain tasks. In particular, lower priority tasks may never get serviced since they would be passed over in the priority scheme. By applying the priority inheritance scheme to the waiting list, a true first-in-first-out scheduling algorithm can be maintained, while also assuring that the executing task is not interrupted.

As per claim 3, Jackson teaches the invention as claimed, including the method in claim 2 wherein:

step (C) further comprises:

4) setting the temporary dispatching priority of the second task to the standard dispatching priority for the second task before substep (2) and (3) (col. 3 lines 54-59, "the computation agent's priority level is restored to its original, i.e., first priority level, when it rejoins the queue to reacquire the lock", wherein the priority level is only changed for a task when it acquires the resource).

As per claim 7, Murata teaches the invention as claimed, including the following limitations not shown by Jackson or McDonald, specifically the method in claim 4 wherein:



step (C) further comprises:

- 4) comparing the temporary dispatching priority for a third task that is ahead of the second task in the first FIFO queue to the temporary dispatching priority of the second task (col. 3 lines 7-15, “the use of the priority inheritance scheme requires modification or rescheduling of the wait queue to respond to the dynamic change of the priority level”, wherein the priority inheritance scheme requires an adjustment in the waiting tasks such that a first-in-first-out dispatching scheme is maintained); and
- 5) setting the temporary dispatching priority of the third task to the temporary dispatching priority of the second task when the temporary dispatching priority of the second task is determined in substep (4) to be greater than the temporary dispatching priority of the third task (col. 3 lines 7-15, “the use of the priority inheritance scheme requires modification or rescheduling of the wait queue to respond to the dynamic change of the priority level”, wherein the priority inheritance scheme allows one task to inherit the priority associated with a subsequent task that has a higher priority).

As per claims 12-13 and 17, Jackson teaches the invention as claimed, including software stored in a computer software storage medium for performing the method of claims 2-3 and 7, respectively (Fig. 1, wherein the invention is disclosed in a computing environment including software for implementing the disclosed method).

7. Claims 4-6, 8, 14-16 and 18 rejected under 35 U.S.C. 103(a) as being unpatentable over Jackson in view of McDonald.

As per claim 4, McDonald teaches the invention as claimed, including the following limitations not shown by Jackson, specifically the method in claim 1 wherein:

the method further comprises:

(C) queuing a second task in a first FIFO queue when the second task attempts to lock the first lock and fails (col. 5 lines 30-37, “if a first task [‘task A’] attempts to lock a resource, and the resource is already held by a second task [‘task B’],...task A is added to the wait list”), wherein:

the FIFO queue comprises a set of tasks waiting to lock the first lock (col. 4 lines 28-37, “wait list 200 contains N task entries, labeled tasks 1 to N. Each task in the list is a task that is not currently executing and is waiting for a resource to be freed”); and

step (C) comprises:

- 1) placing the second task at an end of the first FIFO queue (col. 5 lines 30-37, “if a first task [‘task A’] attempts to lock a resource, and the resource is already held by a second task [‘task B’],...task A is added to the wait list”);
- 2) comparing the temporary dispatching priority for the first task to the temporary dispatching priority of the second task (col. 5 lines 30-37, “the priority of task A is compared against the priority of task B so that the priority of task B may be shifted accordingly”); and
- 3) setting the temporary dispatching priority of the first task to the temporary dispatching priority of the second task when the temporary dispatching priority of the second task is determined in substep (2) to be greater than the temporary

dispatching priority of the first task (col. 5 lines 38-50, "If the priority of task A is higher than the current priority of task B, then task Bs priority is elevated to that of task A, and the current priority of task B in the hold list is updated to reflect the new priority level").

It would have been obvious to one of ordinary skill in the art to combine Jackson and McDonald since the method disclosed by Jackson, while raising the priority of an executing task to avoid a system interrupt, fails to account for the arrival of another task that may be of higher priority. Thus, the disclosure of McDonald, which allows an executing task to inherit the priority of a waiting task that is of a higher priority allows the executing task to finish its work on the shared resource, and thereby ensuring the resource is left in a consistent state upon the executing task's completion.

As per claim 5, McDonald teaches the invention as claimed, including the method in claim 4 wherein step (C) further comprises:

- 4) testing whether the first task is in a second FIFO queue awaiting a chance to lock a second lock (col. 7 lines 54-64, "Fig. 6 is a flowchart that illustrates the steps of locking multiple resources"); and
- 5) upgrading the temporary dispatching priority of a third task that has a second lock locked when the first task is determined in substep (4) to be in the second FIFO queue awaiting a chance to lock the second lock, wherein substep (5) comprises:
  - a) comparing the temporary dispatching priority for the third task to the temporary dispatching priority of the second task (col. 5 lines 30-37, "the priority

of task A is compared against the priority of task B so that the priority of task B may be shifted accordingly”; col. 8 lines 25-28, “The aforementioned principles concerning multiple read/single write functionality and deadlock detection...apply equally to multiple record locking as single record locking”); and

b) setting the temporary dispatching priority of the third task to the temporary dispatching priority of the second task when the temporary dispatching priority of the second task is determined to be greater than the temporary dispatching priority of the third task (col. 5 lines 38-50, “If the priority of task A is higher than the current priority of task B, then task Bs priority is elevated to that of task A, and the current priority of task B in the hold list is updated to reflect the new priority level”; col. 8 lines 25-28, “The aforementioned principles concerning multiple read/single write functionality and deadlock detection...apply equally to multiple record locking as single record locking”).

As per claim 6, McDonald teaches the invention as claimed, including the method in claim 5 wherein:

the method further comprises:

D) comparing a highest temporary dispatching priority of any task in the first FIFO queue to the temporary dispatching priority of the first task after the first task unlocks the second lock (col. 4 lines 28-37, “The tasks entries are queued in the order of priority to

receive the next free resource. Thus, task 1 is of higher priority than task 2, and so on down to task N"); and

E) setting the temporary dispatching priority of the first task to the highest temporary dispatching priority of any task in the first FIFO queue when the temporary dispatching priority of the first task is determined in step (D) to be greater than the highest temporary dispatching priority of any task in the first FIFO queue (col. 5 lines 38-50, "If the priority of task A is higher than the current priority of task B, then task Bs priority is elevated to that of task A, and the current priority of task B in the hold list is updated to reflect the new priority level").

As per claim 8, McDonald teaches the invention as claimed, including the method in claim 1 wherein:

the temporary dispatching priority for the first task is determined dynamically and adjusted dynamically depending on the temporary dispatching priority for other tasks behind the first task in a first FIFO queue associated with the first lock (col. 5 lines 38-50, "If the priority of task A is higher than the current priority of task B, then task Bs priority is elevated to that of task A, and the current priority of task B in the hold list is updated to reflect the new priority level").

As per claims 14-16 and 18, Jackson teaches the invention as claimed, including software stored in a computer software storage medium for performing the method of claims 4-6 and 8, respectively (Fig. 1, wherein the invention is disclosed in a computing environment including software for implementing the disclosed method).

***Conclusion***

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

USPN 6,021,425 to Waldron, III et al. teaches of comparing an incoming task's priority to that of an executing task, and invoking different scheduling algorithms based on the comparison.


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Syed Ali  
March 11, 2004



MENG-AI T. AN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100